UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/824,751	04/15/2004	Patrick H. Dussud	MS1-1962US	7046

22801 7590 11/29/2010

LEE & HAYES, PLLC
601 W. RIVERSIDE AVENUE
SUITE 1400
SPOKANE, WA 99201

EXAMINER	
SAVLA, ARPAN P	

ART UNIT	PAPER NUMBER
2185	

NOTIFICATION DATE	DELIVERY MODE
11/29/2010	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

lhptoms@leehayes.com

Office Action Summary	Application No.	Applicant(s)	
	10/824,751	DUSSUD, PATRICK H.	
	Examiner	Art Unit	
	Arpan P. Savla	2185	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *15 September 2010*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,3,4,6-12,14,17-19,21 and 23-31* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,3,4,6-12,14,17-19,21 and 23-31* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Continued Examination Under 37 CFR 1.114

A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on September 15, 2010 has been entered.

Response to Amendment

This Office action is in response to Applicant's communication filed September 15, 2010 in response to the Office action dated August 4, 2010. Claims 1, 7, 12, and 19 have been amended. Claims 1, 3, 4, 6-12, 14, and 17-19, 21, and 23-31 are pending in this application.

REJECTIONS BASED ON PRIOR ART

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. Claims 1, 3, 4, 6-12, 14, and 17-19, 21, and 23-31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Dussud (U.S. Patent 6,502,111) in view of Applicant's admitted prior art (hereinafter "AAPA").

3. **As per claim 1**, Dussud discloses a computer-readable storage medium apparatus having computer-executable instructions encoded thereon to support ephemeral garbage collection by setting a write-watch mechanism to watch specified memory locations (col. 5, lines 42-45; Fig. 1, element 32), the computer readable storage medium being accessible by a computing device (col. 14, lines 26-35), the instructions when executed, configuring the computer device to perform operations comprising:

during a loop, requesting via the write-watch mechanism a list of memory locations (col. 5, lines 50-51), *It should be noted that the "write watch module" is equivalent to a "write-watch mechanism".*

the list: identifying a plurality of the memory locations that have been accessed since a last ephemeral garbage collection process (col. 5, lines 42-45; col. 6, lines 34-44), each memory location corresponding to one of a plurality of cards associated with one or more objects allocated from with a memory heap, each of the plurality of cards associated with a card table, wherein the card table identifies one or more of the plurality of cards with one or more objects that have been accessed (col. 5, line 64 – col. 6, line 1; Fig. 1, elements 34 and 36a-36i; col. 9, lines 3-9). *It should be noted that each set of "memory objects" (i.e. 36a-36i) is equivalent to a "card" and the "bit map" is equivalent to a "card table".*

comprising a bitmap, wherein each bit within the bitmap corresponds to one of the plurality of cards, modification of the bitmap occurring when a corresponding bit is set at the time that the card is trimmed to disk (col. 5, lines 56-63 and the document incorporated by reference in col. 5, lines 56-63 (i.e. U.S. patent application 09/628,708));

and performing garbage collection upon at least one accessed object (col. 11, lines 42-53; Fig. 4, element 318).

Dussud does not disclose instructions when executed, configuring the computing device such that during execution of a program, when a statement of the program for execution is obtained, the computing device is configured to determine whether the statement includes a store operator;

during a loop, in an event the statement has a store operator:

storing a value specified in the statement in a memory location specified in the statement;

and determining whether the memory location specified is within an ephemeral generation;

in an event the memory location specified is within the ephemeral generation, obtaining a next statement of the program for execution;

in an event the memory location specified is not within the ephemeral generation, setting a card associated with the memory location specified and obtaining the next statement of the program for execution;

creating, during the current ephemeral garbage collection process, a bundle table containing entries identifying a plurality of bundles, wherein each of the plurality of bundles identifies groupings of subsets of the plurality of cards;

marking, outside of the loop including setting the card, during the current ephemeral garbage collection process, two or more of the plurality of bundles identified in the bundle table using the list, wherein the marked bundles identify groupings of subsets of the plurality of marked cards having associated objects that have been accessed since a last ephemeral garbage collection process.

AAPA discloses instructions when executed, configuring the computing device such that during execution of a program, when a statement of the program for execution is obtained, the computing device is configured to determine whether the statement includes a store operator (paragraph 0007; Fig. 1, element 100);

during a loop, in an event the statement has a store operator:

storing a value specified in the statement in a memory location specified in the statement (paragraph 0007; Fig. 1, element 104);

and determining whether the memory location specified is within an ephemeral generation (paragraph 0007; Fig. 1, element 102);

in an event the memory location specified is within the ephemeral generation, obtaining a next statement of the program for execution (paragraph 0007; Fig. 1, element 102);

in an event the memory location specified is not within the ephemeral generation, setting a card associated with the memory location specified and obtaining the next statement of the program for execution (paragraph 0007; Fig. 1, element 106);

creating, during the current ephemeral garbage collection process, a bundle table containing entries identifying a plurality of bundles, wherein each of the plurality of bundles identifies groupings of subsets of the plurality of cards (paragraph 0007); *It should be noted that the "bundle bit map" is equivalent to the "bundle table".*

marking, outside of the loop including setting the card, during the current ephemeral garbage collection process, bundles, wherein the marked bundles identify groupings of subsets of the plurality of marked cards having associated objects that have been accessed since a last ephemeral garbage collection process (paragraph 0007; Fig. 1, element 108); *It should be noted that a "card" which has been "accessed" is equivalent to a "marked card".*

Dussud and AAPA are analogous art because they are from the same field of endeavor, that being garbage collection systems.

At the time of the invention it would have been obvious to a person of ordinary skill in the art to apply AAPA's helper code which performs storage and card and bundle marking to Dussud's concurrent garbage collection, such that the combined garbage collection system would, mark, during the current ephemeral garbage collection process, a plurality of bundles identified in the bundle table of AAPA using Dussud's list. The motivation for doing so would have been to reduce the amount of heap that is analyzed during garbage collection.

4. **As per claim 3**, the combination of Dussud/AAPA discloses the write-watch mechanism operates within a memory manager (Dussud, col. 5, lines 35-36; Fig. 1, elements 28 and 32).

5. **As per claim 4**, the combination of Dussud/AAPA discloses the write-watch mechanism records a first access to a one of the plurality of memory locations (Dussud, col. 5, lines 42-45).

6. **As per claim 6**, the combination of Dussud/AAPA discloses the write-watch mechanism maintains the list of memory locations in response to a request from the ephemeral garbage collection process (Dussud, col. 5, lines 50-51; Fig. 1, elements 30 and 32). *It should be noted that the "garbage collector" is equivalent to a "garbage collection process".*

7. **As per claim 7**, the combination of Dussud/AAPA discloses the operations further comprising resetting the list of memory locations (Dussud, col. 5, lines 52-55).

8. **As per claim 8**, the combination of Dussud/AAPA discloses the subset of cards corresponds to a number of cards that are tracked using a page of memory storing the card table (AAPA, paragraph 0006; Dussud, col. 5, lines 64-67; Fig. 1, element 34).

9. **As per claim 9**, the combination of Dussud/AAPA discloses identifying the marked bundle comprises marking a bit associated with the marked bundle table within a bundle bitmap based on the memory locations within the list (AAPA, paragraph 0007; Fig. 1, element 108; Dussud, col. 5, lines 42-45 and 50-51).

10. **As per claim 10**, the combination of Dussud/AAPA discloses marking the bit comprises setting the bit (AAPA, paragraph 0007).

11. **As per claim 11**, the combination of Dussud/AAPA discloses determining the at least one marked card comprises scanning a card bitmap having a bit for each of the plurality of cards, the bit for each marked card being different than another bit of the card bitmap associated with one of the cards that was not accessed (AAPA, paragraph 0006).

12. **As per claim 12**, Dussud discloses a method for executing statements within a program to support ephemeral garbage collection (col. 8, lines 46-53; Fig. 3) by setting a write-watch mechanism to watch specified memory locations (col. 5, lines 42-45; Fig. 1, element 32), the method comprising:

specifying a range of table memory to watch during program execution by calling a write-watch mechanism that performs tracking of access to table memory; and maintains a write-watch list that identifies cards marked within the table memory since a garbage collection process was last performed (col. 8, line 65 – col. 9, line 11; Fig. 2, element 204; col. 5, lines 42-50), each card being associated with and updated upon access to one or more objects allocated within a memory heap, the memory heap being divided into a plurality of cards, a subset of that plurality of cards that are tracked using a page of card table memory that are tracked using a page of card table memory outside of a loop including marking at least one of the plurality of cards (col. 5, line 64 – col. 6, line 1; Fig. 1, elements 34 and 36a-36i).

Dussud does not disclose the memory heap being divided into a plurality of cards with each card being grouped into one of a plurality of bundles;

during the loop including marking at least one of the plurality of cards, in an event the statement obtained has a store operator:

storing a value within the memory heap at a memory location specified by the statement obtained;

and determining whether the memory location specified is within an ephemeral generation;

in an event the memory location specified is within the ephemeral generation, obtaining a next statement of the program for execution;

in an event the memory location specified is not within the ephemeral generation, marking the at least one of the plurality of cards within the card table memory corresponding to the memory location and obtaining a next statement of the program for execution.

AAPA discloses each card being grouped into one of a plurality of bundles, wherein each of the plurality of bundles corresponds to a subset of that plurality of cards (paragraph 0007);

during the loop including marking at least one of the plurality of cards, in an event the statement obtained has a store operator:

storing a value within the memory heap at a memory location specified by the statement obtained (paragraph 0007; Fig. 1, element 104);

and determining whether the memory location specified is within an ephemeral generation (paragraph 0007; Fig. 1, element 102);

in an event the memory location specified is within the ephemeral generation, obtaining a next statement of the program for execution (paragraph 0007; Fig. 1, element 102);

in an event the memory location specified is not within the ephemeral generation, marking the at least one of the plurality of cards within the card table memory corresponding to the memory location and obtaining a next statement of the program for execution (paragraph 0007; Fig. 1, element 106).

Dussud and AAPA are analogous art because they are from the same field of endeavor, that being garbage collection systems.

At the time of the invention it would have been obvious to a person of ordinary skill in the art to apply AAPA's helper code which performs storage and card and bundle marking to Dussud's concurrent garbage collection, such that the combined garbage collection system would, mark, during the current ephemeral garbage collection process, a plurality of bundles identified in the bundle table of AAPA using Dussud's list. The motivation for doing so would have been to reduce the amount of heap that is analyzed during garbage collection.

13. **As per claim 14**, the combination of Dussud/AAPA discloses the tracking includes the write-watch mechanism that resides within a memory manager and setting bits in the card table upon access to at least one of the plurality of cards (Dussud, col. 5, lines 35-36; Fig. 1, elements 28 and 32; AAPA, paragraph 0006).

14. **As per claim 17**, the combination of Dussud/AAPA discloses the ephemeral garbage collection process requests the list when performing garbage collection (Dussud, col. 5, lines 50-51; Fig. 1, element 30).

15. **As per claim 18**, the combination of Dussud/AAPA discloses the ephemeral garbage collection process determines a marked bundle based on the write-watch list (AAPA, paragraph 0007; Dussud, col. 5, lines 42-45). *See the rejection of claim 12 above. It should be noted that when combining Dussud and AAPA as set forth in the rejection of claim 12 above, any markings to AAPA's bundle table would be based on Dussud's array of memory locations.*

16. **As per claim 19**, Dussud discloses a memory management system configured to set a write-watch mechanism to watch specified memory locations during execution of a program (col. 5, lines 42-45; Fig. 1, element 32), the system comprising:

a processor (col. 7, line 1; Fig. 2, element 104);

and a memory into which a plurality of instructions are loaded (col. 7, lines 35-39; Fig. 2, element 106) and into which a plurality of objects are dynamically allocated, the memory having a heap into which the objects are allocated, the heap being divided into a plurality of cards, each card being associated with one or more of the plurality of objects (col. 5, line 64 – col. 6, line 1; Fig. 1, elements 34 and 36a-36i);

and the write-watch mechanism configured to identify cards that have been set in the memory location specified since a garbage collection process was last performed (col. 8, line 65 – col. 9, line 11; Fig. 2, element 204; col. 5, lines 42-50), and outside the

loop including setting the card (col. 5, line 64 – col. 6, line 1; Fig. 1, elements 34 and 36a-36i).

Dussud does not disclose obtaining a statement of the program for execution, determining whether the statement obtained includes a store operator,

the heap being divided into a plurality of cards which are grouped into a plurality of bundles,

the system being configured to, based at least on whether the store operator is included in the statement for execution obtained, perform an operation such that:

in an event the statement obtained does not have a store operator, executing the statement;

and in an event the statement obtained has a store operator performing a loop including:

storing a value specified in the statement obtained in a memory location specified in the statement obtained;

determining whether the memory location specified is within an ephemeral generation;

in an event the memory location specified is within the ephemeral generation, obtaining a next statement of the program for execution;

in an event the memory location specified is not within the ephemeral generation, setting a card associated with the memory location specified and obtaining the next statement of the program for execution;

and the plurality of cards being grouped into one of the plurality of bundles, and a corresponding bundle of the plurality of bundles that have been marked outside of the loop including setting the card.

AAPA discloses obtaining a statement of the program for execution, determining whether the statement obtained includes a store operator (paragraph 0007; Fig. 1, element 100);

the heap being divided into a plurality of cards which are grouped into a plurality of bundles (paragraph 0007);

the system being configured to, based at least on whether the store operator is included in the statement for execution obtained, perform an operation such that:

in an event the statement obtained does not have a store operator, executing the statement (paragraph 0007; Fig. 1, element 100);

and in an event the statement obtained has a store operator performing a loop including:

storing a value specified in the statement obtained in a memory location specified in the statement obtained (paragraph 0007; Fig. 1, element 104);

determining whether the memory location specified is within an ephemeral generation (paragraph 0007; Fig. 1, element 102);

in an event the memory location specified is within the ephemeral generation, obtaining a next statement of the program for execution (paragraph 0007; Fig. 1, element 102);

in an event the memory location specified is not within the ephemeral generation, setting a card associated with the memory location specified and obtaining the next statement of the program for execution (paragraph 0007; Fig. 1, element 106);

and the plurality of cards being grouped into one of the plurality of bundles (paragraph 0007);

Dussud and AAPA are analogous art because they are from the same field of endeavor, that being garbage collection systems.

At the time of the invention it would have been obvious to a person of ordinary skill in the art to apply AAPA's helper code which performs storage and card and bundle marking to Dussud's concurrent garbage collection, such that the combined garbage collection system would, mark, during the current ephemeral garbage collection process and outside of the loop, a plurality of bundles identified in the bundle table of AAPA using Dussud's list. The motivation for doing so would have been to reduce the amount of heap that is analyzed during garbage collection.

17. **As per claim 21**, the combination of Dussud/AAPA discloses the write-watch mechanism resides within a memory manager and sets bits in the card table upon access to at least one of the plurality of cards (Dussud, col. 5, lines 35-36; Fig. 1, elements 28 and 32; AAPA, paragraph 0006).

18. **As per claim 25**, the combination of Dussud/AAPA discloses the write-watch mechanism sets bits in the card table memory upon access to at least one of the plurality of cards at the time that the card is trimmed to disk (col. 5, lines 56-63 and the document incorporated by reference in col. 5, lines 56-63).

19. **As per claim 26**, the combination of Dussud/AAPA discloses determining whether calling the write-watch mechanism resets a write- watch state by inquiring which cards have changed without being considered as having asked and thereby resetting the state (col. 5, lines 56-63 and the document incorporated by reference in col. 5, lines 56-63);

and in an event the state is to be reset, placing a separate reset call to reset the range of card table memory without reporting whether the cards in the range have been marked (col. 5, lines 56-63 and the document incorporated by reference in col. 5, lines 56-63).

20. **As per claim 27**, the combination of Dussud/AAPA discloses determine whether a request resets a write-watch state by inquiring which memory locations have changed without being considered as having asked and thereby resetting the state (col. 5, lines 56-63 and the document incorporated by reference in col. 5, lines 56-63);

and in an event the state is to be reset, the system being configured to place a separate reset request to reset a range of memory locations without reporting whether the memory locations in the range have been marked (col. 5, lines 56-63 and the document incorporated by reference in col. 5, lines 56-63).

21. **As per claim 28**, the combination of Dussud/AAPA discloses the operations further comprising using, by a current ephemeral garbage collection process, information from the write-watch mechanism to determine which bundles in older generations have objects for collection (AAPA, paragraph 0007; Dussud, col. 11, lines 9-31; Fig. 4, element 314).

22. **As per claim 29**, the combination of Dussud/AAPA discloses the operations further comprising:

for each marked bundle identified in the bundle table, determining at least one marked card in a grouping of subsets of the plurality of marked cards identified by the marked bundle (AAPA, paragraph 0007);

and for each determined marked card, determining at least one accessed object within associated with the marked card (AAPA, paragraph 0007).

As per claim 30, the combination of Dussud/AAPA discloses tracking access to the card table memory by the write-match mechanism (Dussud, col. 8, line 65 – col. 9, line 11; Fig. 3, element 204; col. 5, lines 42-45 50-51; Fig. 1, element 32);

creating, during an ephemeral garbage collection process, one or more bundle tables containing entries identifying groupings of the cards in the plurality of bundles, for each stored statement within the program, the ephemeral garbage collection process occurring after the program execution process (AAPA, paragraph 0007);

updating, during the ephemeral garbage collection process, at least one bundle table by marking the entries in the bundle table based on information obtained from the write-watch list, wherein the updated marked bundle table identifies groupings the plurality of marked cards having associated objects that have been accessed since a last garbage collection process (AAPA, paragraph 0007; Fig. 1, element 108; Dussud, col. 5, lines 42-45 and 50-51);

for each marked bundle table, determining during the ephemeral garbage collection process at least one marked card in a grouping of the plurality of marked cards identified by the marked bundle table (AAPA, paragraph 0007);

for each marked card, determining during the ephemeral garbage collection process at least one accessed object associated with the marked card (AAPA, paragraph 0007);

and performing garbage collection during the ephemeral garbage collection process upon the at least one accessed object (Dussud, col. 11, lines 42-53; Fig. 4, element 318).

23. **As per claim 31**, the combination of Dussud/AAPA discloses the system further being configured to:

request a list from the write-watch mechanism (Dussud, col. 5, lines 50-51; Fig. 1, element 32), the list identifying memory locations that have been written into since a last garbage collection process (Dussud, col. 5, lines 42-45; col. 6, lines 34-44), each memory location corresponding to one of the plurality of cards associated with a card table, wherein the card table identifies one or more cards that have been accessed (Dussud, col. 9, lines 3-9), the card table and cards being marked to identify the one or more of the plurality of cards with the one or more objects that have been accessed, during a first program execution process exclusive of an ephemeral garbage collection process (Dussud, col. 8, line 65 – col. 9, line 11; Fig. 2, element 204; col. 5, lines 42-50);

create, during the ephemeral garbage collection process, one or more bundle tables wherein each bundle table identifies groupings of the plurality of cards in the plurality of bundles (AAPA, paragraph 0007);

update, during the ephemeral garbage collection process, at least one bundle table by marking bundles within the bundle table based on the list, wherein the marked bundles corresponds to marked cards having associated objects that have been accessed since a last garbage collection process (AAPA, paragraph 0007; Fig. 1, element 108; Dussud, col. 5, lines 42-45 and 50-51);

determine, during the current ephemeral garbage collection process, for each marked bundle within the bundle table, at least one marked card within the marked bundle, the at least one marked card indicating that one or more objects associated with the marked card have been accessed (AAPA, paragraph 0007);

determine, during the current ephemeral garbage collection process, for each marked card, the one or more objects that have been accessed (AAPA, paragraph 0007);

and perform, during the current ephemeral garbage collection process, garbage collection upon the one or more accessed objects (Dussud, col. 11, lines 42-53; Fig. 4, element 318).

24. **As per claim 23**, the combination of Dussud/AAPA discloses the marked bundle being identified by a marked bit associated with the marked bundle within a bundle bitmap based on the list (AAPA, paragraph 0007; Dussud, col. 5, lines 42-45). *See the rejection of claim 19 above. It should be noted that when combining Dussud and AAPA*

as set forth in the rejection of claim 19 above, any markings to AAPA's bundle table would be based on Dussud's array of memory locations.

25. **As per claim 24**, the combination of Dussud/AAPA discloses further being configured to set a bit in the card table to identify one or more cards that have been accessed at the time a card that has been accessed is trimmed to disk (Dussud, col. 5, lines 56-63 and the document incorporated by reference in col. 5, lines 56-63).

Response to Arguments

26. Applicant's arguments filed September 15, 2010 with respect to **claims 1, 3, 4, 6-12, 14, and 17-19, 21, and 23-31** have been fully considered but they are not persuasive.

27. With respect to Applicant's arguments regarding claims 1, 12, and 19, the Examiner respectfully disagrees. The Examiner notes that when taking the broadest reasonable interpretation of the term "loop", it follows that a loop can have only a single iteration. In the case where a loop has only a single iteration, all the steps within the loop are performed only once each, with control then passing it steps outside the loop. Thus, in such a case it does not matter whether a step is inside or outside the loop, because any given step is performed only once. Now turning to the current claim language, in the case where the claimed loop in claims 1, 12, and 19 has only a single iteration, all steps within the claimed loop are performed only once each. Thus, it follows that any step within those claims, whether inside or outside the claimed loop, is performed only once. Therefore, in the case where the claimed loop in claims 1, 12,

and 19 has only a single iteration, the combination of Dussud/AAPA still reads on said claims. Accordingly, the combination of Dussud/AAPA renders claims 1, 12, and 19 unpatentable.

28. As for Applicant's arguments with respect to the dependent claims, the arguments rely on the allegation that the independent claims are patentable and therefore for the same reasons the dependent claims are patentable. However, as addressed above, the independent claims are not patentable, thus, Applicant's arguments with respect to the dependent claims are not persuasive.

Conclusion

STATUS OF CLAIMS IN THE APPLICATION

The following is a summary of the treatment and status of all claims in the application as recommended by MPEP 707.70(i):

CLAIMS REJECTED IN THE APPLICATION

Per the instant office action, **claims 1, 3, 4, 6-12, 14, and 17-19, 21, and 23-31** have received an action on the merits and are subject of a non-final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Arpan P. Savla whose telephone number is (571) 272-1077. The examiner can normally be reached on M-F 8:30-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Sanjiv Shah can be reached on (571) 272-4098. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Arpan P. Savla/
Examiner, Art Unit 2185
November 22, 2010